

CHANNEL MERGING METHOD FOR VOD SYSTEM

FIELD OF INVENTION

The present invention relates to a channel merging method, and more particularly to an optimized multicast delivery to a plurality of clients by using the merging of a plurality of channels delivering a video stream in a communication network.

BACKGROUND OF THE INVENTION ART

With the explosive growth of the Internet and the increasing power of computers, interest has grown in a class of application called video-on-demand, where clients can request media files (video, audio, data ..., etc.) at any time for immediate or future watching. However, video-on-demand poses a new challenge, that is, a huge consumption of server bandwidth and network bandwidth. Traditionally, each request is served by a dedicated unicast stream, and the cost of the unicast based VOD system is enormous. The advent of channel merging technology creates a brand new model for VOD service, and its goal is to reduce the server bandwidth required to satisfy clients requesting a particular object video by having them simultaneously receive two or more streams. As clients receive and store the data for immediate watching purposes, the server can have one video object served to more than one user simultaneously by multicast and thus reduce both the network bandwidth and server bandwidth.

Existent channel merging methods can be classified as three types: static broadcast, merge tree construction and event driven. The static broadcast, exemplified by Skyscraper, broadcasts segments of a demanded object in several channels with a specified period and length. The advantage of the static broadcast is its simplicity and relatively high efficiency in very busy environment. However, the performance of the static broadcast is poor when the load of system is not high or the popularity of different objects is disperse due to its rigid resource allocation. The merge tree construction, exemplified by Dyadic, dynamically constructs a merge tree when the new users arrive, with the nodes of tree representing channels. A channel is not allocated until it is really needed by a user. This method overcomes the drawbacks of the static broadcast by eliminating the waste of idle channel resources. However, as the merge tree is exclusively

1 determined by the joining time of new users, it does not directly support VCR-like functions,
2 i.e., random stop, pause, fast/back forward, etc. The Event-driven, exemplified by SRMT (Simple
3 Reachable Merge Target) and CT (Closest Target), dynamically determines a set of channels that
4 the client should subscribe to when the client indicates to the server of playing, stopping,
5 jumping or merging events. VCR-like functions are supported by this method because the merge
6 path for each client is dynamically adjusted according to user interactions.

7 A method of merging of two channels will be described below in conjunction with Figs. 1,
8 2 and 3:

9 At step 1, the VOD server 1 receives a request for playing a video program from a client A
10 and, according to the request, sends the requested video program to the client A on the channel
11 S6.

12 At step 2, when receiving the same VOD request as that of the client A after some time (T)
13 from a client B, the VOD server 1 creates a channel S11 and informs the client B to get ready for
14 receiving from the VOD server 1 the video program on the channel S11 and the channel S6.

15 At step 3, the VOD server 1 sends the video program from its starting point (a) to the
16 client B on the channel S11, and the client B receives it, and meantime the client B receives on
17 the channel S6 in synchronism with the client A and stores the subsequent part of the video
18 program continuously sent from the VOD server 1.

19 At step 4, the VOD server 1 takes the channel S6 as the parent channel of the channel S11
20 (i.e., the channel to which the channel S11 will merge). When the video program received by the
21 client B on the channel S11 reaches the beginning point (b) of the video program that it receives
22 on the channel S6 and stores, i.e., when another time of T is passed, the channel S11 is merged
23 into the channel S6. The VOD server 1 will close the channel S11 and stops sending the video
24 program to the client B on the channel S11. At this time, in the client B is stored the video
25 program from the point (b) to a point (c). After the channel S11 (sub-video stream) is merged
26 into its parent channel S6, if no other client is using the sub-channel S11 (i.e., the sub-channel of
27 the channel S11), then the sub-channel S11 will be terminated.

28 At step 5, after the channel S11 is merged into the channel S6, while the client B continues
29 to receive, on the channel S6, and stores the subsequent part of the video program sent from the
30 VOD server 1 from the point (c), it reads from the point (b) and plays back the video program

1 stored in its local memory in a FIFO manner, enabling the playback of the video program on the
2 client B to be continued.

3 Although the event-driven method for channel merging are the most flexible method for
4 controlling multicast channels, existing methods of this type have an evident drawback. If a
5 channel is removed when it has merged into its parent channel or is canceled due to stopping or
6 jumping events, those clients subscribing the sub-channels of this removed channel have to
7 change the channels they have subscribed. For example, the CT scheme simply chooses the
8 latest video stream channel in the earlier video streams still in the system as the next target to be
9 merged, and the merge target computed by CT are not always reachable, even if no further new
10 sub-channel is created. The reason is that the target stream channel may itself merge with its
11 target channel before it can be reached by later channels. In this case, later stream channel must
12 select a new merge target again by using the CT algorithm. Furthermore, the operations of the
13 target stream channel such as random stopping, pausing, fast-forwarding, etc. will also make it
14 impossible for the later stream channels to merge and will force them to reselect their new parent
15 channels.

16 In order to inform affected clients of the change of merge tree, the video server must
17 actively send a notification to the each of these clients. This could bring about the following
18 disadvantageous effects:

- 19 1. Reverse notifications from a video server to clients significantly may increase the load
20 of the video server, since the number of notifications is proportional to the number of
21 affected clients and the frequency of unexpected channel stopping events.
- 22 2. Clients must be ready to accept incoming connections from unknown regions of the
23 Internet, which increases the possibility for clients to be affected unexpectedly.
- 24 3. The reverse notifications may not be able to pass through the firewall with certain
25 configurations. For example, if a client within a firewall tries to watch a video clip
26 stored in a video server outside the firewall, the server will never be able to initiate the
27 transmission of a notification to the client.

28 **Summary of the Invention**

1 In order to solve the above-mentioned problems, by using a criterion for deciding a merge
2 path in response to events of starting, jumping, merging, and stopping, the present invention
3 provides channel merging methods, apparatus and systems for a VOD system. An example
4 method comprises the steps of: (1) in response to requests from a plurality of clients for a video
5 program, establishing a root channel (S1) and at least one sub-channel (S11), said root channel
6 (S1) being established according to a request from a client that makes the earliest request, each
7 of said sub-channels (S11) being established corresponding to a request of a client that makes a
8 later request; (2) monitoring variation of the number of the clients that are using each of said
9 established channels, and maintaining the channel if the number of the clients using the
10 monitored channel is not zero, and closing the channel if the number of the clients using the
11 monitored channel becomes zero.

12 The present invention also provides a channel merging apparatus for a VOD system, said
13 channel merging apparatus is disposed in a video server in said VOD system or connected to the
14 same operatively, said channel merging apparatus comprises: a channel selecting unit for
15 establishing a root channel (S1) and at least one sub-channel (S11) in response to requests from
16 a plurality of clients for a video program, said root channel (S1) being established according to a
17 request from a client that makes the earliest request, each of said sub-channels (S11) being
18 established in response to a request from a client that makes a later request; a channel control
19 unit for monitoring variation of the number of the clients that are using each of said established
20 channels, and maintaining the channel if the number of the clients using the monitored channel
21 is not zero, and closing the channel if the number of the clients using the monitored channel
22 becomes zero.

23 In the present invention, all the "channel merging events" proceed in the direction from the
24 lowest-level sub-channel to the root channel, therefore no case that the channel which a client is
25 using is removed will occur. Even if a stopping event occurs directly, a channel will not be
26 removed until all clients that use it (in the form of a sub-channel of the channel) explicitly
27 release it. Therefore reverse notifications are avoided, and one client's behavior will not affect
28 other clients, and the load of both the VOD server and the network are reduced.

29 **Brief Description of the Drawings**

1 The above-mentioned advantages and other features of the present invention will become
2 more apparent from the following detailed description in conjunction with the drawings, in
3 which:

4 Fig. 1 shows a multicast network having a VOD server and a plurality of clients.

5 Fig. 2 is a schematic diagram of a video program stream that performs channel merging.

6 Fig. 3 is a schematic diagram of a channel merge tree.

7 Fig. 4 is a timing diagram of request/response occurring between the VOD server and the
8 client of Fig. 1.

9 Fig. 5 is a diagram showing the structure of a channel merging apparatus configured in the
10 VOD server according to the present invention.

11 Fig. 6 is flow chart of the process of the VOD server in the case of occurrence of a
12 “starting event” according to the present invention.

13 Fig. 7 is flow chart of the process of the VOD server in the case of occurrence of a
14 “jumping event” according to the present invention.

15 Fig. 8 is flow chart of the process of the VOD server in the case of occurrence of a
16 “merging event” according to the present invention.

17 Fig. 9 is flow chart of the process of the VOD server in the case of occurrence of a
18 “stopping event” according to the present invention.

19 **Description of the Invention**

20 The present invention provides methods, systems and apparatus to solve the
21 above-mentioned problems, by using a criterion for deciding a merge path in response to events
22 of starting, jumping, merging, and stopping. In an example embodiment, the present invention
23 provides a channel merging method for a VOD system. The method comprises the steps of: (1)
24 in response to requests from a plurality of clients for a video program, establishing a root
25 channel (S1) and at least one sub-channel (S11), said root channel (S1) being established
26 according to a request from a client that makes the earliest request, each of said sub-channels
27 (S11) being established corresponding to a request of a client that makes a later request; (2)

1 monitoring variation of the number of the clients that are using each of said established
2 channels, and maintaining the channel if the number of the clients using the monitored channel
3 is not zero, and closing the channel if the number of the clients using the monitored channel
4 becomes zero.

5 Also provided is a channel merging apparatus for a VOD system. The channel merging
6 apparatus is disposed in a video server in said VOD system or connected to the same
7 operatively. In an example embodiment, the channel merging apparatus comprises: a channel
8 selecting unit for establishing a root channel (S1) and at least one sub-channel (S11) in response
9 to requests from a plurality of clients for a video program, said root channel (S1) being
10 established according to a request from a client that makes the earliest request, each of said
11 sub-channels (S11) being established in response to a request from a client that makes a later
12 request; a channel control unit for monitoring variation of the number of the clients that are
13 using each of said established channels, and maintaining the channel if the number of the clients
14 using the monitored channel is not zero, and closing the channel if the number of the clients
15 using the monitored channel becomes zero.

16 In the present invention, all the "channel merging events" proceed in the direction from the
17 lowest-level sub-channel to the root channel, therefore no case that the channel which a client is
18 using is removed will occur. Even if a stopping event occurs directly, a channel will not be
19 removed until all clients that use it (in the form of a sub-channel of the channel) explicitly
20 release it. Therefore reverse notifications are avoided, and one client's behavior will not affect
21 other clients, and the load of both the VOD server and the network are reduced. Meantime,
22 VCR-like functions, e.g., playing, stopping, seeking (fast/back forward), etc. are inherently
23 supported by this method because the merge tree is dynamically constructed and adjusted when
24 client-initiated events occur, and the merge path for each client is dynamically adjusted based on
25 the merge tree.

26 Another main advantage of the invention is that the method for controlling the VOD server
27 is compatible with the request/response mode of HTTP, thus can be easily implemented upon
28 HTTP. HTTP is an important protocol for exchanging application data on the Internet, and the
29 advent of Web services further consolidates the trend that Internet-based applications should rely

1 on HTTP as much as possible for data transmission. In addition, most firewalls are reluctant to
2 let general traffic other than HTTP pass through.

3 The present invention will be described in detail hereinafter in conjunction with the
4 drawings. Unless noted specially, all such operations as starting, stopping, jumping, merging
5 and etc. mentioned in the following is performed with respect to channels on which the same
6 video program (object) are played.

7 Fig. 1 shows a multicast network having a sever 1 and a plurality of clients 4. The
8 multicast network comprises a VOD server 1, the Internet 2, a firewall 3 and a plurality of
9 clients 4, wherein the VOD server 1 and the clients 4 communicate with each other through the
10 Internet 2 and the firewall 3. In the present invention, the client 4 sends requests to the VOD
11 server 1 through the firewall 3 and the Internet 2, for performing the operations of VCR-like
12 functions such as playing, stopping, pausing, fast/back forwarding on a video clip.

13 The request/response operation occurring between the VOD server 1 and the clients 4 of
14 Fig. 1 will be described below in conjunction with Fig. 4.

15 Fig. 4 is a timing diagram of the request/response occurring between the VOD server 1 and
16 the clients 4 of Fig. 1.

17 In Fig. 4, each message contains the message type of the action to be performed, which is
18 specified by a list of parameters. Here are currently five message types defined: OPEN, PLAY,
19 PAUSE, MERGE and CLOSE. For each message sent by the client, the sever will send back a
20 RESPONSE message.

21 At step (1), by sending an OPEN message to the VOD server 1, the client 4 creates a
22 session with the VOD server 1, the OPEN message containing a video ID for uniquely
23 identifying the requested video file (video program) on the VOD server 1. If the session is
24 successfully created, then the VOD server 1 will send back to the client 4 a RESPONSE
25 message containing channel information, such as multicast address and port number, and the
26 client 4 can receive the requested video program according to the channel information.

27 At step (2), the client 4 sends a PLAY message to request to start playing the video
28 program or resume from paused state. An offset can be specified as a parameter in the PLAY
29 message to search a specified position of the video program. The RESPONSE message may

1 contain additional information for the client 4, such as the information that indicates the client 4
2 to receive the requested program, and etc.

3 At step (3), when the client 4 detects the merging of channels, it sends a MERGE message
4 to the VOD server 1 (of course, the VOD server 1 can also derive the occurrence of a channel
5 merging event by using its components such as the channel control unit 20 to compute the
6 channel). The VOD server 1 will close channels that are not used and send back a RESPONSE
7 message indicating additional channels to which the client 4 should join.

8 At step (4), the client 4 can send a PAUSE message to the VOD server 1 to temporarily
9 halt data transmission while playing the video program, and the VOD server 1 makes a
10 corresponding response.

11 At step (5), the client 4 can send a CLOSE message to the VOD server 1 to explicitly close
12 the session with the VOD server 1, and the VOD server 1 makes a corresponding response.

13 If we model all the above-mentioned requests and notification messages into various
14 events, then there are four types of event pertinent to the VOD server 1, i.e., "starting event",
15 "jumping event", "stopping event" and "merging event". All these events can be operated
16 through the above-mentioned five types of messages, i.e., OPEN, PLAY, PAUSE, MERGE and
17 CLOSE.

18 When the client 4 sends a request for an object (video program), a starting event happens,
19 i.e., the client 4 requests to play a video clip at time "t" (using a PLAY message). A jumping
20 event happens when the client 4 sends a fast-forward or back-forward request for an object. That
21 is, the client 4 sends a request for playing the video program from time "t+s" or "t-s", where "s"
22 is an offset time of an object (i.e., other parts of the requested video program) to be jumped with
23 respect to time "t". At this time, the VOD server 1 creates a new channel to play the video
24 program from time t-s (using the PLAY message), and meantime closes the channel that plays
25 the video program from time "t" (using the CLOSE message, as will be described in detail
26 below). A stopping event happens when the client 4 no longer needs an object, i.e., the channel
27 on which the video program is played is closed (using a CLOSE message). A merging event
28 happens when the client 4 has reached a merge point where the last channel pair the client 4 was
29 watching have successfully merged (using the MERGE message and the CLOSE message), such

1 as the case of the channel merging method described in the portion of the background art of the
2 present invention, i.e., the beginning point (b) in Fig. 2 is the merged point.

3 In order to implement the operations of VCR-like functions such as playing, stopping,
4 pausing, fast forward/backward, the present invention provides a channel merging apparatus 40.

5 The specific structure of the channel merging apparatus 40 according to the present
6 invention will be described now in conjunction with Fig. 5.

7 Fig. 5 is a view showing the structure of the channel merging apparatus 40 according to the
8 present invention.

9 The channel merging apparatus 40 according to the invention is disposed inside the VOD
10 server 1, comprises: a channel selecting unit 10 for receiving requests for a certain video
11 program from a plurality of clients, and for creating a root channel (S1) and at least one
12 sub-channel (S11) in response to the requests, said root channel (S1) being created according to
13 a request of a client that makes the earliest request, each of said plurality of sub-channels (S11)
14 being created in response to the request of a client that makes a request later, the channel
15 selecting unit 10 further seeking a parent channel satisfying with conditional expressions (1) and
16 (2) to be described later for a sub-channel in the channel merging process; and a channel control
17 unit 20 for performing such operations as the creating, merging and closing of channels
18 according to the request of the client 4 and the selection result of the channel selecting unit 10.

19 In addition, the above-mentioned channel merging apparatus 40 can be connected to the
20 VOD server 1 operatively, instead of being disposed inside the VOD server 1. Meantime, the
21 channel selecting unit 10 and the channel control unit 20 can be a single unit, such as a CPU in a
22 computer, for executing an executable program stored in ROM or RAM or other storage
23 medium (not shown) in the computer to effect the functions corresponding to the channel
24 selecting unit 10 and the channel control unit 20.

25 The channel control unit 20 further comprises a count unit 22, which marks the number of
26 the clients 4 using each channel with a count parameter (ref_num) to effect the control function
27 of the channel control unit 20. When a merging, jumping or stopping event happens to said each
28 channel and its sub-channels, the count unit 22 decreases the value of the count parameter, and if
29 the value of the count parameter is zero, the channel whose value of the count parameter is zero
30 is closed at the side of the VOD server 1.

1 If the value of the count parameter is not zero, then the channel is maintained at the VOD
2 server 1, and the client having performed the merging, jumping or stopping event no longer
3 receives the program played on the channel.

4 In the present invention, the channel selecting unit 10 in the VOD server 1 creates a root
5 channel (S1) and at least one sub-channel (S11) in response to the requests for a certain video
6 program from a plurality of clients as shown in Fig. 3, said root channel (S1) being created
7 according to the request of the client that makes the earliest request (such as client A), each of
8 the a plurality of sub-channels (S11) (and also S5 and S6, for example) being created in response
9 to the request of clients that make requests later, and the root channel S1 and the plurality of
10 sub-channels S11 (and S5 and S6, etc.) form a tree structure. Certainly, all of the
11 above-mentioned requests are those satisfying the VOD condition.

12 All the above-mentioned channels transfer multicast streams from the VOD server 1 based
13 on the request of the client 4, and the multicast streams of each channel can be received by all
14 the clients 4. Each client 4 receives at most two channels at the same time, of which one is
15 initiated for the client 4 itself and another one was initiated for a previous client for example the
16 first client requesting the object only receives a video program from a channel s1 (as shown in
17 Fig. 3), while the second client receives the video program from channels s2 and s1
18 simultaneously, in which s1 is selected as the parent channel of s2 (as shown in Fig. 3). All
19 clients must be capable of storing the received video program on a local storage (not shown).
20 For each created channel, the VOD server 1 monitors the variation of the number of the clients 4
21 using the channel through the channel control unit 20 of the present invention. If the number of
22 the clients 4 using the monitored channel is not zero, the channel is maintained; if the number of
23 the clients 4 using the monitored channel is zero, i.e., no client 4 is using the channel, then the
24 VOD server 1 closes the channel through the channel control unit 20 of present invention.

25 Here, the channel control unit 20 in the VOD server 1 detects the merging of two channels
26 (video streams) either by its own calculation or by receiving messages from the client 4 where
27 channel merging occurs. And the merging process continues until all sub-channels have merged
28 into their root channel (the first channel created for the same video program). It is assumed that
29 the length of the video program is infinite here.

1 The channel merging method according to the present invention will be described below
2 with reference to the drawings.

3 Referring to Fig. 3 again, Fig. 3 is a schematic diagram of a channel merge tree. Each node
4 represents a channel, where the first channel created for a certain video program is defined as a
5 root channel. A higher level channel is defined as the parent channel of a lower level, and in
6 contrast the lower level is defined as the sub-channel of the higher level channel. As shown in
7 Fig. 3, S1 is the root channel, S6 is the parent channel of S11, while S11 is the sub-channel of
8 S6, and channels S6, S10, S11 and S12 are a collection of the posterity channels of S5.

9 If the channel control unit 20 in the VOD server 1 creates a new channel, which we
10 assume as channel S11, under the request of the client 4 (starting or jumping of a channel), then
11 the channel selecting unit 10 immediately seeks the parent channel S6 for the channel S11. If the
12 parent channel S6 is found, then the parent channel 6 is returned; otherwise, a message of “can
13 not find the parent channel” is returned.

14 The method executing the above-mentioned operations is as follows:

15 Step 1: by using the following expression (1), find a nearest root channel such as S1 from
16 the collection of active root channels to which the channel S11 will possibly merge, that is:

$$17 \quad \min(S11.start_time - S1.start_time) < object_length/2 \quad (1)$$

18 wherein which, $\min(S11.start_time - S1.start_time)$ indicates a minimal value among the
19 difference values between the start time ($S11.start_time$) of the channel S11 and the start time
20 ($S1.start_time$) of each root channel in the collection of the root channels, and $object_length/2$
21 indicates a half of total length (total time) of the played video program, and wherein the start
22 time ($start_time$) indicates the start time of this channel. Therefore, the above expression means
23 that the minimal value among the difference values between the start time of the channel S11
24 and the start time of a certain root channel S1 in the collection of the root channels is less than
25 the 1/2 total length (total time) of the played video program. In this case, the root channel S1 is
26 considered to be reachable by its sub-channel S11 and can act as the root channel S1 of the
27 channel S11; otherwise, or if the difference value between the start times is larger than the 1/2
28 total length (total time) of the played video program, this root channel is considered to be
29 unreachable, and therefore a message of “can not find the root channel” is returned, with the
30 channel S11 being taken as a new root channel.

1 If an effective root channel S1 is found by using the above-mentioned condition, the
2 method proceeds to the following step 2.

3 Step 2: if the collection of the posterity channels of the root channel S1 is defined as S, the
4 posterity channels described here include the sub-channel S5 whose direct parent channel is root
5 channel S1, and sub-channel S6 of the channel S5. Here the channel S11 is defined as the
6 sub-channel of the channel S6, and meantime it can be known that the channel S11 is also a
7 posterity channel of the channel S5, the rest being deduced by analogy. By using the following
8 expression (2), the channel S6 is found in the collection S (wherein the channel S6 is the parent
9 channel of the channel S11, and both channels S6 and S11 are within the posterity channel
10 collection S and are posterity channels of the root channel S1):

$$11 \quad \min(S11.start_time - S6.start_time) < S6.start_time - S5.start_time \quad (2)$$

12 The above expression means that a parent channel S6 to which the channel S11 will merge
13 is found for the channel S11, and this parent channel S6 should satisfy the following condition:
14 the minimal value among the difference values between the start time of the channel S11 and the
15 start time of the candidate parent channel S6 is less than the difference value between the start
16 time of the parent channel S6 and the start time of the parent channel S5 of the parent channel
17 S6.

18 If there exists a channel S6 satisfying the above-mentioned condition, then the channel S6
19 is returned as the parent channel of the channel S11 in the next merging. Otherwise, or if no
20 channel satisfying the above condition can be found, the root channel S1 is returned as the
21 parent channel of the channel S11.

22 Obviously, if only the above condition is satisfied, it is ensured that the parent channel S6
23 will not merge into the parent channel S5 of the parent channel S6 before the channel S11
24 merges into its parent channel S6.

25 That is to say, at the time a channel S6 merges into its parent channel S5, each sub-channel
26 of the channel S6, such as S10 and S11, has merged into its parent channel S6. Therefore, from
27 the time then knowing that the channel S6 merged into its parent channel S5 by calculation or by
28 receiving the message from the client 4, the VOD server 1 closes the channel S6 so that the
29 usage of other clients is not influenced. This is because if the selection of the channel S6
30 satisfies the above-mentioned condition, no other client is using the channel S6 at this time, i.e.,

1 channels S10 and S11 satisfying the above-mentioned condition have both merged into the
2 channel S6. Thus, it is ensured that all the channel merging starts in a direction from the lowest
3 sub-channel to their root level. That is, the channel will not be removed until all the clients 4
4 using a certain channel explicitly release the channel (in the form of the sub-channel of the
5 channel).

6 The operations of the VOD server 1 and its units therein in response to such 4 types of
7 events as starting, jumping, merging and stopping will be described below in conjunction with
8 Figs. 6, 7, 8 and 9.

9 First, the operations of the VOD server 1 and its units therein in response to a starting
10 event will be described.

11 As shown in Fig. 6, when the client 4 initiates a starting event at time "t" (i.e., the client
12 selects to play a video clip):

13 At step 100, the channel selecting unit 10 creates a new channel S11 to play back the video
14 program, and sets in the channel S11 start_time=t, object_offset=0 (an object offset time
15 indicating the offset time when the channel starts, i.e., the offset of the start time of the channel
16 with respect to the channel starting at time "t") and ref_num=1 (reference value indicating the
17 number of the clients), where start_time=t indicates the channel starts at time "t",
18 object_offset=0 indicates the channel has no offset, and ref_num=1 indicates only one client is
19 using the channel.

20 At step 102, the parent channel of the channel S11 is sought by the channel selecting unit
21 10.

22 At step 103, it is determined whether or not there exists the parent channel.

23 At step 104, if no parent channel is found, then the channel control unit 20 takes the
24 channel S11 as a new root channel (i.e., sets root_flag=1 (a root mark indicating whether the
25 channel is a root channel)), and this channel S11 is the only channel that the client 4 should
26 watch.

27 Otherwise, at step 105, if the parent channel S6 is found, then the client 4 must watch the
28 channel S11 and its parent channel S6 simultaneously.

1 At step 106, the above operation information is sent to the client 4 in response to the
2 starting request initiated by the client 4.

3 Next, the operations of the VOD server 1 and its units therein in response to a jumping
4 event will be described.

5 As shown in Fig. 7, when the client 4 initiates a jumping event at the object offset time "s"
6 with respect to time "t" (i.e., the client 4 performs VCR-like functions such as fast
7 forward/backward at time "t-s"):

8 At step 200, the channel selecting unit 10 creates a new channel S11 to play back the video
9 program starting from time "t-s", and sets in the channel S11 start_time=t-s,
10 object_offset=s and ref_num=1, where start_time=t-s indicates the channel starts from time
11 "t-s", object_offset=s indicates the offset time of the start time of the channel with respect to the
12 channel starting from time "t" is "s", and ref_num=1 indicates only one client is using the
13 channel.

14 At step 202, if the channel starting at time "t" is S4, then the value of the count parameter
15 of the channel S4 is decreased, i.e., because the channel S4 previously used by the client 4 has
16 been caused to perform a stopping operation, the number of the clients using the channel S4 is
17 currently decreased by 1. At this time, if the count parameter of the channel S4 is zero,
18 indicating that no client is using the channel S4 currently, then the channel control unit 20 closes
19 the channel S4; and in contrast, if at this time the count parameter of the channel S4 is not zero,
20 indicating that there are still some clients 4 using the channel S4, then the channel S4 can not be
21 closed, so as to be used continuously by other clients using the channel S4 (such as S8 and S9).
22 But the client 4 having invoked the jumping event no longer uses the channel S4. Instead, it
23 turns to the channel S11 to watch the video program.

24 At step 203, the parent channel of the channel S11 is sought by the channel selecting unit
25 10.

26 At step 204, it is determined whether there exists the parent channel.

27 At step 205, if no parent channel is found, then the channel control unit 20 takes the
28 channel S11 as a new root channel (i.e., sets root_flag=1), and this channel S11 is the only
29 channel that the client 4 should watch.

1 Otherwise, at step 206, if the parent channel S6 is found, then the client 4 must watch the
2 channel S11 and its parent channel S6 simultaneously.

3 At step 208, the above operation information is sent to the client 4 in response to the
4 starting request initiated by the client 4.

5 Next, the operations of the VOD server 1 and its units therein in response to a merging
6 event will be described.

7 As shown in Fig. 8, when the client 4 initiates a merging event (i.e., the merging of the
8 sub-channel S11 into the parent channel S6 occurs):

9 At step 300, the channel control unit 20 decreases the value of the count parameter of the
10 sub-channel S11. That is, since the channel S11 used by the client 4 has merged into its parent
11 channel S6, the number of the clients using the channel S11 is currently decreased by 1. At this
12 time, if the count parameter of the channel S11 is zero, indicating that no client is using the
13 channel S11 currently, then the VOD server 1 closes the channel S11; in contrast, if at this time
14 the count parameter of the channel S11 is not zero, indicating that some clients are still using the
15 channel S11, then the channel S11 can not be closed, so that other clients can use the channel
16 S11 continuously. However, the client 4 initiating the merging event no longer uses the channel
17 S11, and jumps to the channel S6 to watch the video program. The value of the count parameter
18 of the channel S11 is the number of clients using the channel S11. For example, if the count
19 parameter is 1 (i.e., ref_num=1), it is indicated that one client 4 is using the channel S11. If the
20 count parameter is 5 (i.e., ref_num=5), it is indicated that five clients 4 are using the channel, all
21 of these channels being the posterity channels of the channel S11.

22 At step 302, the parent channel of the channel S6 is sought by the channel selecting unit
23 10.

24 At step 303, it is determined whether or not there exists the parent channel.

25 At step 304, if no parent channel of the channel S6 is found, then the channel S6 is taken
26 as a new root channel (i.e., sets root_flag=1), and this channel S6 is the only channel that the
27 client 4 should watch.

28 Otherwise, at step 305, if the parent channel S5 of the channel S6 is found, the client 4
29 must watch the channel S6 and its newly determined parent channel S5 simultaneously.

1 At step 306, the above operation information is sent to the client 4 to respond to the
2 merging request initiated by the client 4.

3 Next, the operations of the VOD server 1 and its units therein in response to a stopping
4 event will be described.

5 As shown in Fig. 9, when the client 4 initiates a stopping event (i.e., the client 4 performs a
6 stopping operation of VCR-like functions), or an object (video program) reaches an ending point
7 (i.e., termination):

8 Assuming that a client is using the channel S11 to watch a video program, then at step 400,
9 it is determined whether or not the video program reaches an ending point, i.e. whether the video
10 program has terminated.

11 If the object (video program) reaches an ending point, then the channel control unit 20, at
12 step 402, closes the channel S11 being used by the client 4, and directly releases all the resources
13 of the channel.

14 If the video program does not reach the ending point, then at step 404, the channel control
15 unit 20, as described above, decreases the value of the reference value of the channel S11. If the
16 reference value of the channel S11 is zero, then the channel S11 is closed and all the resources
17 of the channel are released; and in contrast, if the reference value of the channel S11 is not zero,
18 then the channel control unit 20 does not close the channel S11, so that other clients can use the
19 channel S11 continuously, and the client having invoked the "stopping event" no longer uses the
20 channel S11.

21 The above-mentioned method for controlling the VOD server of the present invention is
22 compatible with the request/response mode of HTTP, thus can be easily implemented upon
23 HTTP. HTTP is an important protocol for exchanging application data on the Internet, and the
24 advent of Web services further consolidates the trend that Internet-based applications should rely
25 on HTTP as much as possible for data transmission. In addition, most firewalls are reluctant to
26 let general traffic pass through, but there is not this problem for the HTTP.

27 Variations described for the present invention can be realized in any combination desirable
28 for each particular application. Thus particular limitations, and/or embodiment enhancements
29 described herein, which may have particular advantages to a particular application need not be

1 used for all applications. Also, not all limitations need be implemented in methods, systems
2 and/or apparatus including one or more concepts of the present invention.

3 The present invention can be realized in hardware, software, or a combination of hardware
4 and software. A visualization tool according to the present invention can be realized in a
5 centralized fashion in one computer system, or in a distributed fashion where different elements
6 are spread across several interconnected computer systems. Any kind of computer system - or
7 other apparatus adapted for carrying out the methods and/or functions described herein - is
8 suitable. A typical combination of hardware and software could be a general purpose computer
9 system with a computer program that, when being loaded and executed, controls the computer
10 system such that it carries out the methods described herein. The present invention can also be
11 embedded in a computer program product, which comprises all the features enabling the
12 implementation of the methods described herein, and which - when loaded in a computer system
13 - is able to carry out these methods.

14 Computer program means or computer program in the present context include any
15 expression, in any language, code or notation, of a set of instructions intended to cause a system
16 having an information processing capability to perform a particular function either directly or
17 after conversion to another language, code or notation, and/or reproduction in a different
18 material form.

19 Thus the invention includes an article of manufacture which comprises a computer usable
20 medium having computer readable program code means embodied therein for causing a function
21 described above. The computer readable program code means in the article of manufacture
22 comprises computer readable program code means for causing a computer to effect the steps of a
23 method of this invention. Similarly, the present invention may be implemented as a computer
24 program product comprising a computer usable medium having computer readable program
25 code means embodied therein for causing a a function described above. The computer readable
26 program code means in the computer program product comprising computer readable program
27 code means for causing a computer to effect one or more functions of this invention.

28 Furthermore, the present invention may be implemented as a program storage device readable by
29 machine, tangibly embodying a program of instructions executable by the machine to perform
30 method steps for causing one or more functions of this invention.

1 It is noted that the foregoing has outlined some of the more pertinent objects and
2 embodiments of the present invention. This invention may be used for many applications.
3 Thus, although the description is made for particular arrangements and methods, the intent and
4 concept of the invention is suitable and applicable to other arrangements and applications. It
5 will be clear to those skilled in the art that modifications to the disclosed embodiments can be
6 effected without departing from the spirit and scope of the invention. The described
7 embodiments ought to be construed to be merely illustrative of some of the more prominent
8 features and applications of the invention. Other beneficial results can be realized by applying
9 the disclosed invention in a different manner or modifying the invention in ways known to those
10 familiar with the art.

11 The present invention has been described in detail in the above. It will be understood by
12 those skilled in the art that various changes to the present invention according to the spirit and
13 teaching concept of the present invention will all fall into the scope defined by the appended
14 claims of the present invention.